

【Komutlar】

Bölüm 1 PLC Ladder Diyagram ve Mnemonic Kodlama Kuralları

Bu bölümde, ladder diyagramın temel prensipleri, mnemonic kodlama kuralları, FB-O7C gibi programlama aracının nasıl kullanıldığı anlatılacaktır. Eğer, mnemonic kodlama kurallarına ve PLC ladder diyagramına aşinaysanız bu bölümü geçebilirsiniz.

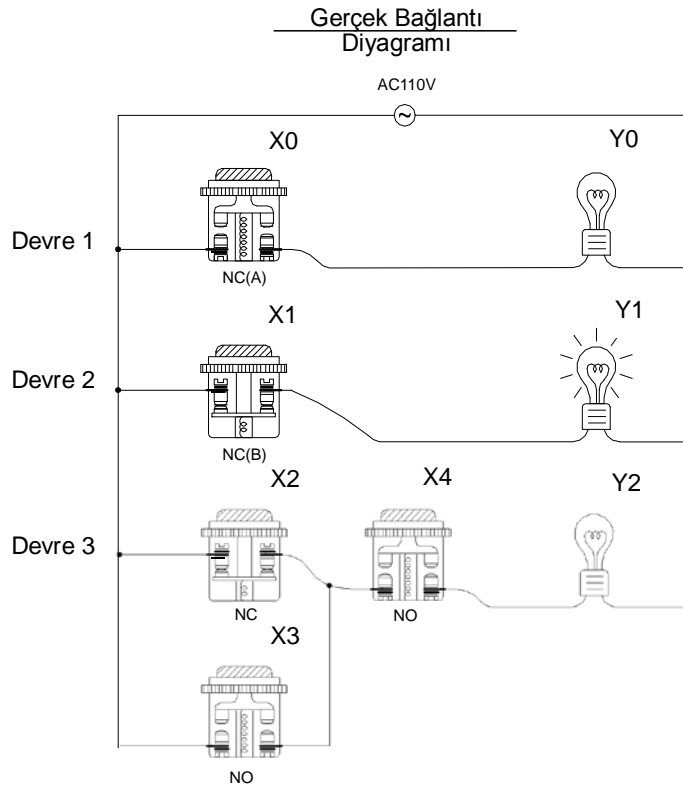
1.1 Ladder Diyagramın Çalışma Prensibi

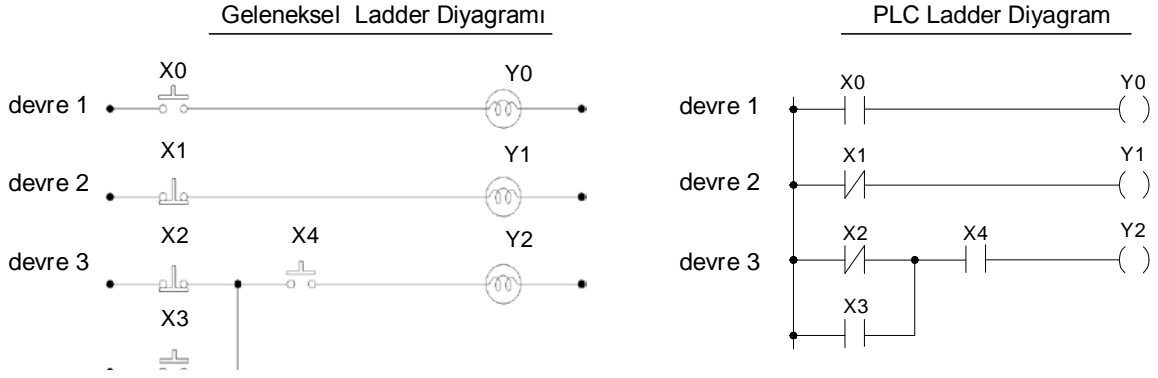
Ladder Diyagram, 2. dünya savaşından bu yana otomatik kontrol sistemleri için kullanılan bir grafik dilidir. Bugüne kadar, otomatik kontrol sistemleri için kullanılan en eski ve en popüler dildir. Başlangıçta, sayıcı, zamanlayıcı, çıkış sargısı, B kontağı (normalde Kapalı) ve A kontağı (normalde Açık) gibi bir kaç basit eleman vardı. Geleneksel sistemler difransiyel kontaklar, kalıcı bobinler (sayfa 1-6'ya bakın) ve diğer komutları sağlayamazlar.

Geleneksel ve PLC Ladder Diyagramının her ikisinin de temel çalışma prensipleri aynıdır. İki sistem arasındaki temel fark PLC ler için semboller bilgisayar ekranı için basitleştirilmiş durumda iken, geleneksel Ladder Diyagram için sembollerin görünümü gerçek cihazlara daha yakındır. Ladder Lojik Diyagram için uygun iki tip lojik sistem vardır. Bunlar, kombinasyon lojik ve sıralı lojiktir. Bu iki lojik için detaylı açıklama aşağıda anlatılmıştır.

1.1.1 Kombinasyon Lojik

Ladder Diyagramın kombinasyon lojigi ; seri veya paralel bağlı, bir veya daha fazla giriş elemanı içeren ve bobinlerin, Zamanlayıcılar/Sayıcılar in sonuçlarını çıkış elemanlarına gönderen bir devredir.

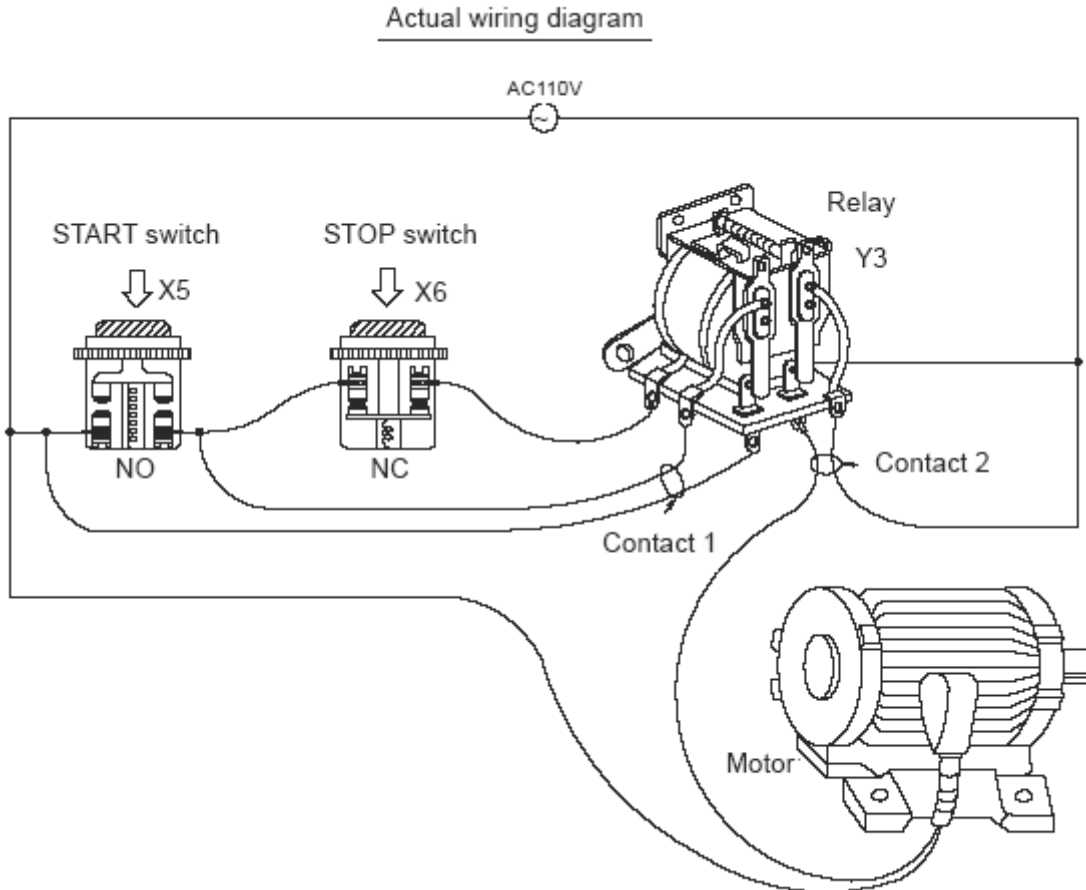




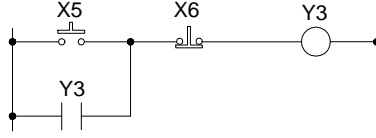
Kombinasyon lojik kullanılarak yapılan Geleneksel Ladder Diyagram ve PLC Ladder Diyagram üstteki şekilde gösterilmiştir. Devre 1 NO (normalde Açık) bir anahtar kullanılmaktadır bu "A" anahtarı veya kontağı şeklinde de adlandırılmıştır. Normal durumda (anahtara basılmamışken), anahtar kontağı OFF durumunda ve ışık sönmüştür. Eğer anahtara basıldıysa kontağın durumu ON durumuna gelir ve ışık yanar. Buna karşılık, devre 2'de kullanılan NC (Normalde kapalı) anahtar "B" anahtarı veya kontağı şeklinde adlandırılmıştır. Normal durumda, Anahtar kontağı ON durumunda ve ışık yanmaktadır. Eğer anahtara basıldıysa, kontağın durumu OFF konumunda olacak ve ışık sönmük durumda olacaktır. Devre 3 bir giriş elemanından fazla eleman içermektedir. Y2 çıkış ışığının yanması için X2 kontağının kapalı veya X3 kontağının ON konumuna gelmesi ve X4 kontağının mutlaka ON konumunda olması gerekmektedir.

1.1.2 Ardışık Lojik

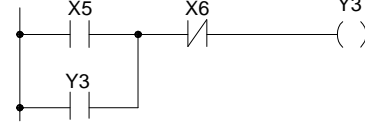
Ardışık lojik geri besleme kontrollü bir devredir; Bu devrenin çıkışı geri besleme ile devreye bir giriş gibi tekrardan verilecektir. Giriş durumu orjinal pozisyona dönse de, çıkış sonuçları aynı durumda kalır. Bu işlem aşağıda gösterildiği gibi kilitlenmiş bir motor sürücüsünün ON/OFF devresi tarafından en iyi şekilde açıklanabilmektedir.



Geleneksel Ladder Diyagram



PLC Ladder Diyagram



Güç kaynağı bu devreye ilk kez bağlandığında, X6 anahtarı ON ama X5 anahtarı OFF olur, bu yüzden Y3 rölesi OFF olur. 1 ve 2 numaralı röle çıkış kontakları OFF'dur. Çünkü onlar A kontağına (anahtarlama ON olduğunda ON olur) aittirler. Motor çalışmaz. Eğer X5 anahtarına basılırsa hem röle ON, hem de 2 numaralı kontaklar ON olur ve motor çalışır. Anahtarlama bir kez açıldığında, X5 anahtarına basılmasada (OFF'a dönerler), anahtarlama kontak 1'den geri besleme desteği ile durumunu korur ve bu Latch (tutucu) devresi olarak adlandırılır. Aşağıda, şekilde gösterilen uygulamanın anahtarlama prosesi ifade edilmiştir.

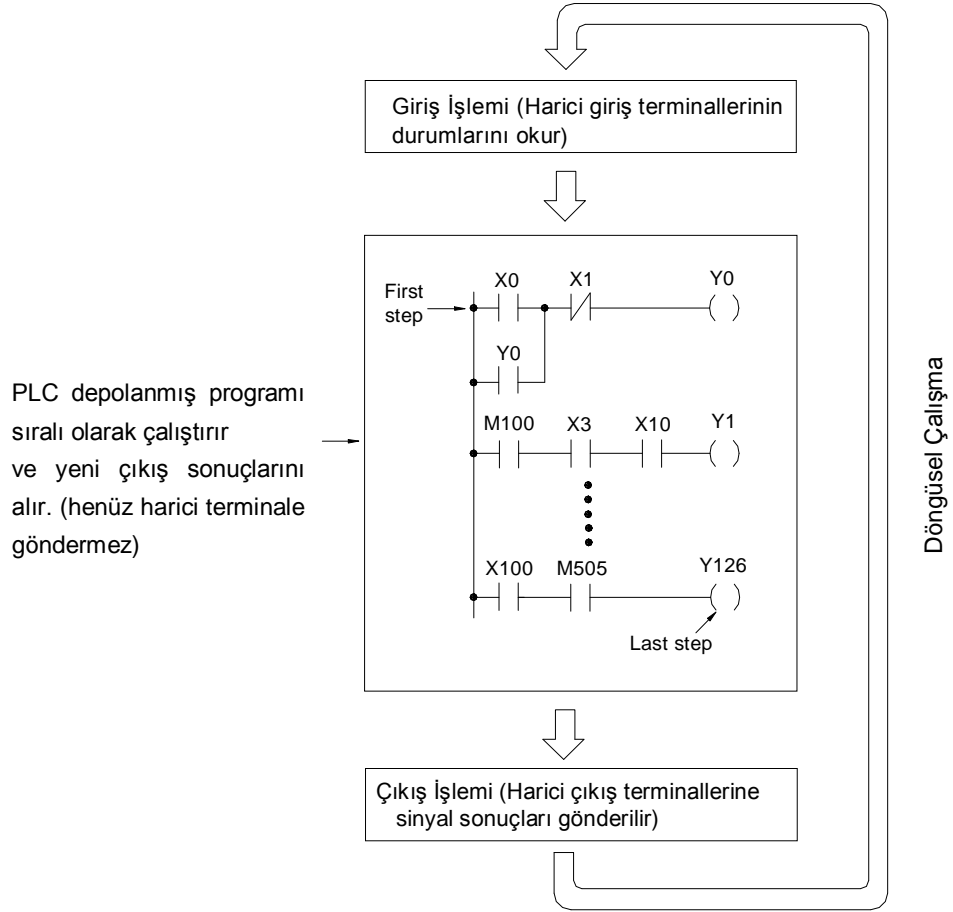
	X5 switch (NO)	X6 switch (NC)	Motor (Relay) status
1	Basılmamış	Basılmamış	OFF
↓			
2	Basılmış	Basılmamış	ON
↓			
3	Basılmamış	Basılmamış	ON
↓			
4	Basılmamış	Basılmış	OFF
↓			
5	Basılmamış	Basılmamış	OFF

Tablodan sıranın farklı durumlarına bakabilirsiniz, sonuçlar giriş durumları aynı olsa bile farklıdır. Örneğin, durum 1 ve durum 3 e bakalım X5 ve X6 anahtarları her ikisinde de basılı değildir, ama motor durum 3 de çalışırken durum 1de durmaktadır. Çıkışın girişe geri besleme ile sıralı kontrolü Ladder Diyagram devresinin unique karakterleridir. Bazen Ladder Diyagramı "Dizi Kontrol Devresi" ve PLC "sıralayıcı" şeklinde de adlandırabiliriz. Bu bölümde, sadece örnekteki gibi çıkış bobinleri ve A/B kontakları kullanılır. Sıralı komutların daha fazla detayı için bölüm 5 - "Sıralı komutlara giriş" bölümüne bakınız.

1.2 Geleneksel Ladder Diyagram ve PLC Ladder Diyagram Arasındaki Farklar

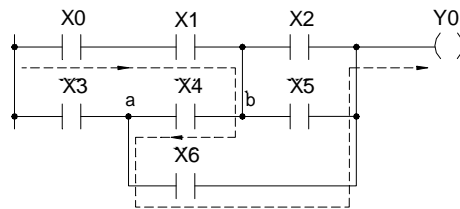
Geleneksel Ladder diyagramı ve PLC Ladder Diyagramının temel çalışma prensipleri birbirine benzerdir. PLC'de kullanılan CPU geleneksel Ladder Diyagram çalışmasına benzemektedir. PLC kullanılan tarama yöntemi ile çıkış bobinleri ve giriş elemanlarını görüntüler, Ladder diyagramın sonuçları geleneksel Ladder Diyagram Lojik Çalışması tarafından üretilen sonuçlar ile aynıdır. Sadece bir CPU bulunmaktadır, bu yüzden PLC sırayı incelemekte ve ilk basamaktan son basamağa doğru çalıştırmaktadır sonra çalışmayı tekrarlayarak ilk basamağa geri dönmektedir (döngüsel çalışma). Çalışmasının tek bir döngüsü tarama zamanı şeklinde adlandırılır. Tarama zamanı program boyutuna göre değişmektedir. Eğer tarama zamanı çok uzunsa, giriş ve çıkışta gecikme yaşanacaktır. Daha uzun gecikme zamanı sürekli çalışan sistemlerin kontrol edilmesinde büyük sorunlara sebep olabilir. Böyle durumlarda kısa tarama zamanına sahip PLC'ler gerekmektedir. Bu yüzden, tarama zamanı PLC'ler için önemli bir özelliktir. Bugünlerde tarama hızı ASIC teknolojileri ve mikro bilgisayarlardaki gelişmelerden dolayı büyük bir gelişme içerisinde. Tipik bir FB_E-PLC kontakların IK adımları için yaklaşık olarak 0.33ms sürmektedir.

PLC Ladder Diyagramının tarama işlemi aşağıdaki şekilde gösterilmektedir.

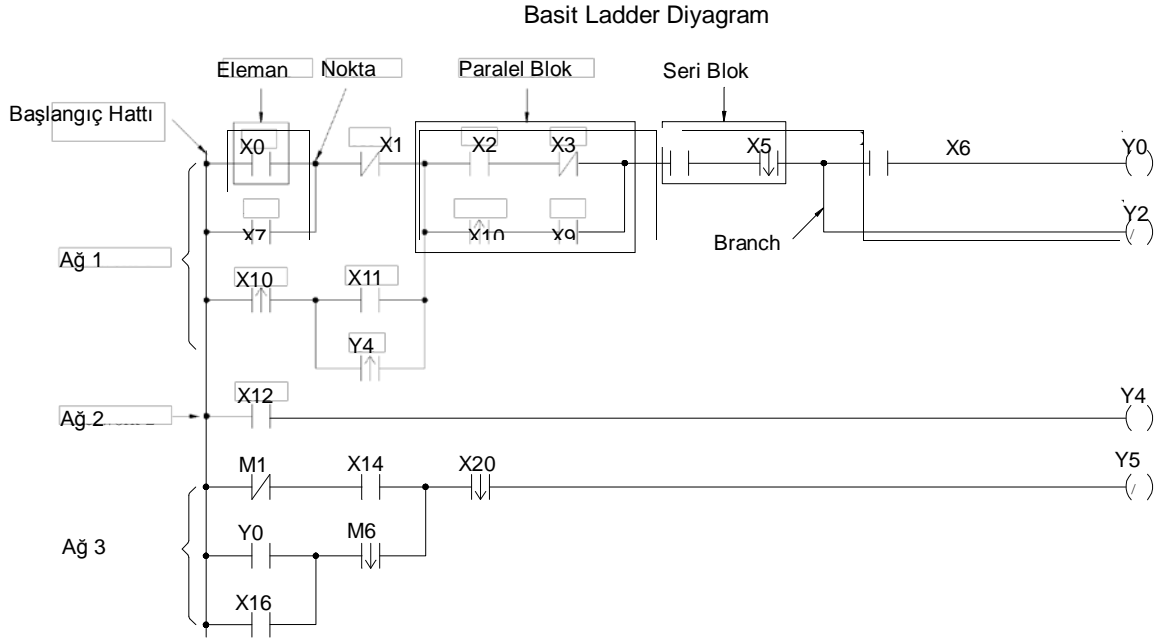


Tarama zamanı farkının dışında, geleneksel ve PLC Ladder Diyagram arasındaki diğer fark "Ters Akış" karakteristiğidir. Aşağıdaki diyagramda gösterildiği gibi, eğer X0, X1, X4 ve X6 ON ise geride kalan elemanlar OFF'dur. Geleneksel Ladder Diyagram devresindeki, Y0 çıkışı için ters akış yolu kesik çizgili hat ile tanımlanmış ve Y0 ON olmaktadır. PLC için ise Y0 OFF'dur, çünkü soldan sağa doğru PLC Ladder Diyagram taraması, eğer X3 OFF sonra CPU "a" noktası OFF ise, PLC önce X3'ü taradıktan sonra X4 ve "b" noktasını ON yapar. Geleneksel Ladder iki yönlü akabilemektirken PLC Ladder sadece soldan sağa doğru ilerleyebilmektedir.

Geleneksel Ladder Diyagramın Ters Akışı



1.3 Ladder Diyagram Yapısı ve Teknik Terimler



(Açıklama: FBS-PLC ağının maksimum boyutu 16 satır x 22 sütün)

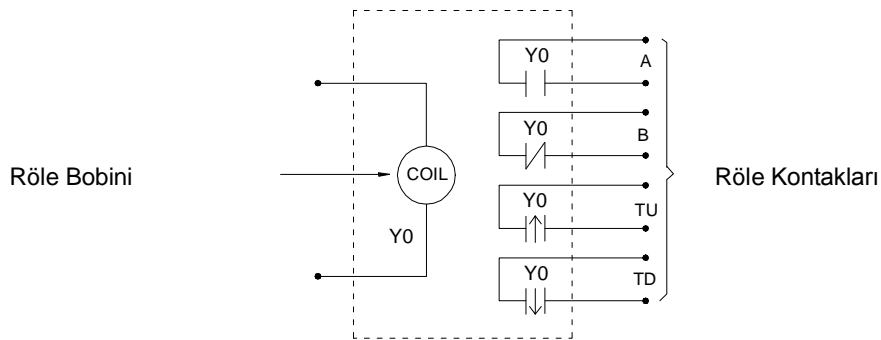
Üstte gösterilen Ladder diyagram çok küçük hücelere bölünmüştür. Bu örnekteki Ladder Diyagram için toplam 88 adet hücre (8 sıra, 11 sütun) bulunmaktadır. Tek hücreye bir eleman yerleştirilebilmektedir. Tamamlanmış bir Ladder Diyagram spesifik gereksinimlere göre tüm hücrelerle birlikte bağlanarak biçimlendirilebilmektedir. Ladder Diyagramla ilişkili teknik terimler aşağıda gösterilmiştir.

1- Kontak

Kontak, açık veya kapalı durumlu bir elemandır. Kontakın bir çeşidi "giriş kontağı" (X ile başlayan bir referans numarası) ve harici sinyallerden durum referansı (giriş terminal bloğundan gelen giriş sinyali) olarak adlandırılmıştır. Diğerleri ise, röle bobininin durumlarını yansıtan "röle kontağı" olarak adlandırılır. (2 başvurunuz). Referans numarası ve kontak durumları arasındaki ilişki kontak tipine bağlıdır. FBS serisi tarafından desteklenen kontak elemanları şunları içerir: A kontağı, B kontağı, yükselen/düşen kenar (TU/TD) kontakları ve Açık/kapalı kontaklar. Daha fazla detay için 4'e bakınız.

2- Röle

Aynı geleneksel röledeki gibi bir kontak ve bir bobin içermektedir.

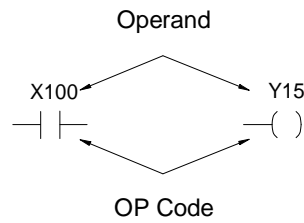


Rölenin çalışması için ilk önce röle bobini enerjilendirilmelidir. Bobin enerjilendikten sonra kontak durumu da On olacaktır. Yukarıdaki şekilde görüldüğü gibi, eğer Y0 çalışırsa, A röle kontağı ON ve B kontağı OFF olduktan sonra sadece 1 tarama süresi için TU kontağı ON ve TD kontağı OFF olur. Eğer Y0 OFF konumuna döndürülürse A röle kontağı ON ve B kontağı OFF olduktan sonra TU kontağı OFF ve TD kontağı sadece bir tarama süresi için ON konumuna dönecektir. (A,B,TU ve TD kontaklarının çalışmaları için "Sıralı Komutlara Giriş" Bölüm 5'e bakınız).

FBs – PLC de dört tip röle vardır, Bunlar; $Y\Delta\Delta\Delta$ (çıkış rölesi), $M\Delta\Delta\Delta\Delta$ (iç röle), $S\Delta\Delta\Delta$ (step röle) ve $TR\Delta\Delta$ (geçici röle) şeklinde adlandırılmışlardır. Çıkış rölesinin durumları çıkış terminal bloğuna gönderilecektir.

③ Başlangıç Hattı: Ladder Diyagramın sol tarafından başlayan hattır.

④ Eleman: Elemanlar Ladder Diyagramın temel birimleridir. Bir eleman alttaki diyagramda gösterildiği gibi iki parçadan oluşmaktadır. Eleman sembollerine "OP Code" ve Referans numaralarına "Operand" denmektedir.



Eleman Tipi	Sembol	Mnemonic Komutlar	Açıklama
A Kontakı (Normalde AÇIK)	$\square\Delta\Delta\Delta\Delta$ — —	(ORG, LD, AND, OR) $\square\Delta\Delta\Delta\Delta$	\square X, Y, M, S, T, C (bölüm 3.2'ye bakınız)
B Kontakı (Normalde KAPALI)	$\square\Delta\Delta\Delta\Delta$ — /—	(ORG, LD, AND, OR) NOT $\square\Delta\Delta\Delta\Delta$	
Yükselen Kenar	$\square\Delta\Delta\Delta\Delta$ — ↑—	(ORG, LD, AND, OR) TU $\square\Delta\Delta\Delta\Delta$	\square X, Y, M, S olabilir
Düşen Kenar	$\square\Delta\Delta\Delta\Delta$ — ↓—	(ORG, LD, AND, OR) TD $\square\Delta\Delta\Delta\Delta$	
Açık Devre Kontakı	—○—	(ORG, LD, AND, OR) OPEN	
Kısa Devre Kontakı	●—●	(ORG, LD, AND, OR) SHORT	
Çıkış Bobini	$\square\Delta\Delta\Delta\Delta$ —()	OUT $\square\Delta\Delta\Delta\Delta$	\square Y, M, S olabilir
Tersleyen Çıkış Bobini	$\square\Delta\Delta\Delta\Delta$ —(/)	OUT NOT $\square\Delta\Delta\Delta\Delta$	
Tutucu(Latching) Çıkış Bobini	$Y\Delta\Delta\Delta$ —(L)	OUT L $Y\Delta\Delta\Delta$	

Açıklama : X,Y,M,S,T ve C kontaklarının aralıkları için bölüm 3.2'ye bakınız. X,Y,M,S,T ve C kontaklarının karakteristikleri için bölüm. 2'ye bakınız.

Ladder Diyagramda gösterilmeyen OUT TRn, LD TRn ve FOn olarak adlandırılan üç tane özel sıra komutu vardır. "Fonksiyon Çıkışı FO" için bölüm 5.1.4'e ve "Geçici röle kullanımı" için bölüm 1,6'ya bakınız.

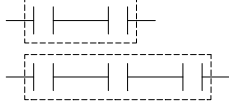
5- Nokta: İki veya daha fazla eleman arasındaki bağlantı noktalarıdır (bölüm 5.3'e bakınız)

6- Blok: Bir devre iki veya daha fazla eleman içerir.

İki tip blok vardır:

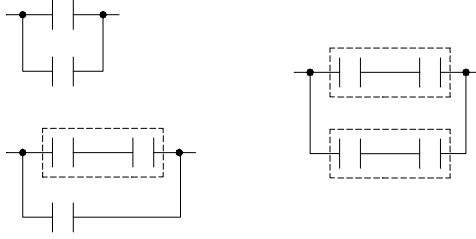
- Seri blok : İki veya daha fazla eleman devrede tek bir sırada seri bağlanmışlardır.

Örnek:



- Paralel blok: Paralel blok, seri blokların veya elemanların paralel bağlanarak oluşturdukları paralel bir kapalı devre tipidir.

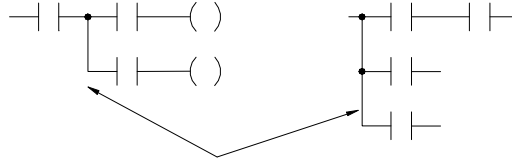
Örnek:



Açıklama: Seri veya paralel şekilde tamamlanan blok tek elemanın kombinasyonu ile biçimlendirilebilir. Mnemonic girişle bir Ladder Diyagram tasarlandığında, eleman, seri ve paralel bloklar içindeki devreleri analiz etmek gerekmektedir. Bölüm 1.5'e bakınız.

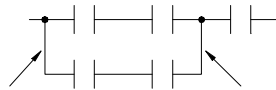
7- Dallar: Herhangi bir ağda, elde edilmektedir. Eğer dikey hattın sağ tarafındaysa devrenin iki veya daha fazla sırası bağlanmıştır.

Örnek:



Dallar

Birleştirilen hat, bir dal hattının sağ tarafında başka bir dikey hat gibi tanımlanabilir. Bu dallı devreler kapalı bir devre içindedir. (paralel blok formu gibi). Bu dikey hat "Birleştirilmiş Hat" şeklinde adlandırılır.

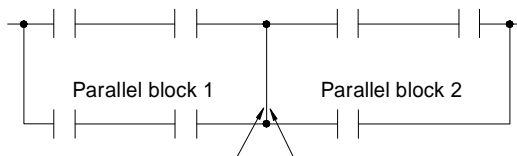


Dallı line

Birleştirilmiş line

Eğer dikey hattın sağ ve sol alanlarının her ikisi de devrenin iki veya daha fazla sırası ile bağlanmışsa bu durumda alttaki şekilde gösterildiği gibi bir birleştirilmiş hat ve bir dallı hat oluşmaktadır.

Örnek:



Block 1 merge line Block 2 branch line

8 - Ağ: Ağ, fonksiyonu belirlenmiş bir devreyi temsil etmektedir. Elemanlar, dallar ve bloklardan oluşur. Ağ, Ladder diyagramdaki tamamlanmış fonksiyonları çalıştırabilen temel bir birimdir. Ladder Diyagramının programı ağ ile birlikte bağlanarak biçimlendirilmiştir. Ağ'ın başlangıcı başlangıç hattıdır. Eğer iki devre dikey bir hatla bağlanmışsa aynı ağa aittirler. Eğer iki devre arasında dikey bir hat yoksa bu devreler iki farklı ağa aittirler. Şekil 1'de üç (1 ~ 3) ağ gösterilmiştir.

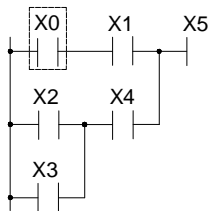
1.4 Mnemonic Kodlama Kuralları

(WinProladder kullanabiliyorsanız bu bölümü geçebilirsiniz)

WinProladder yazılım paketiyle FB-PLC programlamak çok kolaydır. Ladder sembollerini anında bir ladder diyagram programından direk olarak CRT ekranınızda görebilirsiniz. Ama kullanıcılar için FB-PLC programlamak FBC-07 kullanarak kendileri mnemonic komutlar içinde ladder diyagrama dönüştürmektedirler. Çünkü FPC-07 sadece mnemonic komutlu giriş programıdır. Bu bölüm, 1.6'ya kadar mnemonic komutlar içinde ladder diyagramları dönüştüren kodlama kurallarını anlatacaktır.

- Program düzeltme talimatları, soldan sağa ve üstten altta doğrudur. Bu yüzden ağın başlama noktası ağın sol köşesi üstünde olmalıdır. Giriş kontrolü olmadan fonksiyon talimatları hariç, bir ağın ilk talimatı ORG ile başlamalıdır ve ORG talimatı her ağda kullanılmaktadır. Diğer açıklamalar için bölüm 6.1.1'e bakınız.

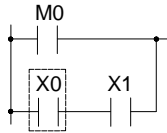
Bölüm:



ORG	X	0
AND	X	1
LD	X	2
OR	X	3
AND	X	4
ORLD		
AND	X	5

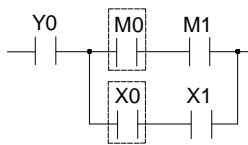
- Ağın başlangıcı hariç (başlama hattı ya dal hattı) dikey hattın bağlanması için LD komutu kullanılır.

Örnek 1:



ORG	M	0
LD	X	0
AND	X	1
ORLD		

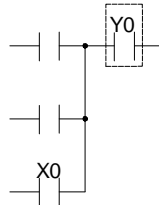
Örnek 2:



AND	Y	0
LD	M	0
AND	M	1
LD	X	0
AND	X	1
ORLD		

Açıklama 1: Eğer elemanların tek satırı dal hattına seri olarak bağlanmışsa AND komutu direk olarak kullanılır.

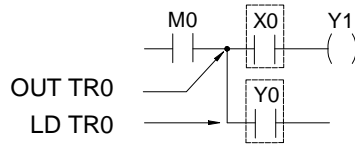
Örnek:



AND	X	0
ORLD		
AND	Y	0

Açıklama 2: OUT TR komutu, noktaların durumlarını depolamakta dal hattı kullanırsa. AND komutu direk olarak kullanılır.

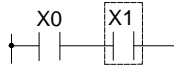
Örnek:



AND	M	0
OUT TR	0	
AND	X	0
OUT	Y	1
LD TR	0	
AND	Y	0

- Tek bir elemanın seri bağlanması için AND komutu kullanılır.

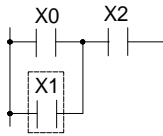
Örnek:



ORG	X	0
AND	X	1

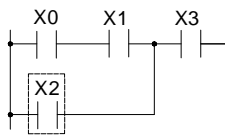
- Tek bir elemanın paralel bağlanması için OR komutu kullanılır.

Örnek:



ORG	X	0
OR	X	1
AND	X	2

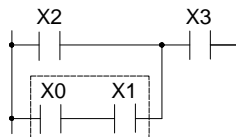
Örnek:



ORG	X	0
AND	X	1
OR	X	2
AND	X	3

- Eğer paralel eleman seri bağlanırsa mutlaka ORLD komutu kullanılmalıdır.

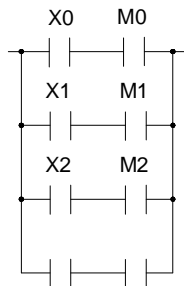
Örnek:



ORG	X	2
LD	X	0
AND	X	1
ORLD		
AND	X	3

Açıklama : Eğer paralel bağlı ikiden fazla blok varsa , üstteki dizi alttaki diziye bağlanmalıdır. Örneğin; blok 1 ve blok 2 ilk önce bağlanmıştır daha sonra blok 3 bağlanmıştır.

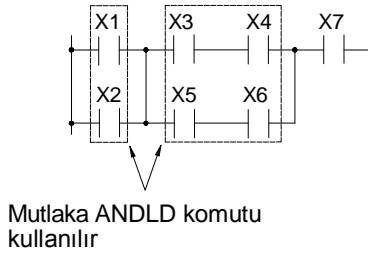
Örnek:



LD	X	0
AND	M	0
LD	X	1
AND	M	1
ORLD		
LD	X	2
AND	M	2
ORLD		
LD	X	3
AND	M	3

- ANDLD komutu paralel blokları seri bağlamak için kullanılır.

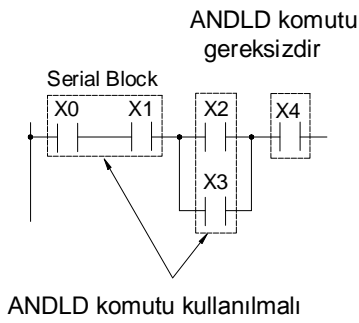
Örnek:



ORG	X	1
OR	X	2
LD	X	3
AND	X	4
LD	X	5
AND	X	6
ORLD		
ANDLD		
AND	X	7

- ANDLD komutu, eleman veya seri blok paralel bloğun önündeyse mutlaka kullanılmalıdır. Paralel blok eleman veya seri bloğun önündeyse, AND komutu tüm diğer parçalarla birlikte bağlanmasında kullanılır.

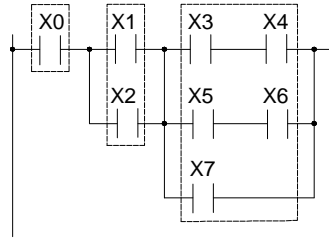
Örnek:



ORG	X	0
AND	X	1
LD	X	2
OR	X	3
ANDLD		
AND	X	4

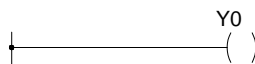
Açıklama: Eğer seri olarak bağlanmış iki bloktan fazla blok varsa üstteki dizi alttaki diziye bağlanmalıdır. Örneğin; blok1 ve blok 2 bağlanmış sonra blok 3 bağlanmıştır.

Örnek:



ORG	X	0
LD	X	1
OR	X	2
ANDLD		
LD	X	3
AND	X	4
LD	X	5
AND	X	6
ORLD		
OR	X	7
ANDLD		

- Çıkış bobini komutu (OUT) sadece ağın sonuna yerleştirilmiş olmalı ve diğer elemanlar olmadan en son bağlanmalıdır. Çıkış bobini direk olarak başlangıç hattına bağlanamaz. Eğer başlangıç hattına çıkış bobini bağlamak istersek kısa devre kontağı ile seri olarak bağlayabiliriz.



ORG SHORT
OUT

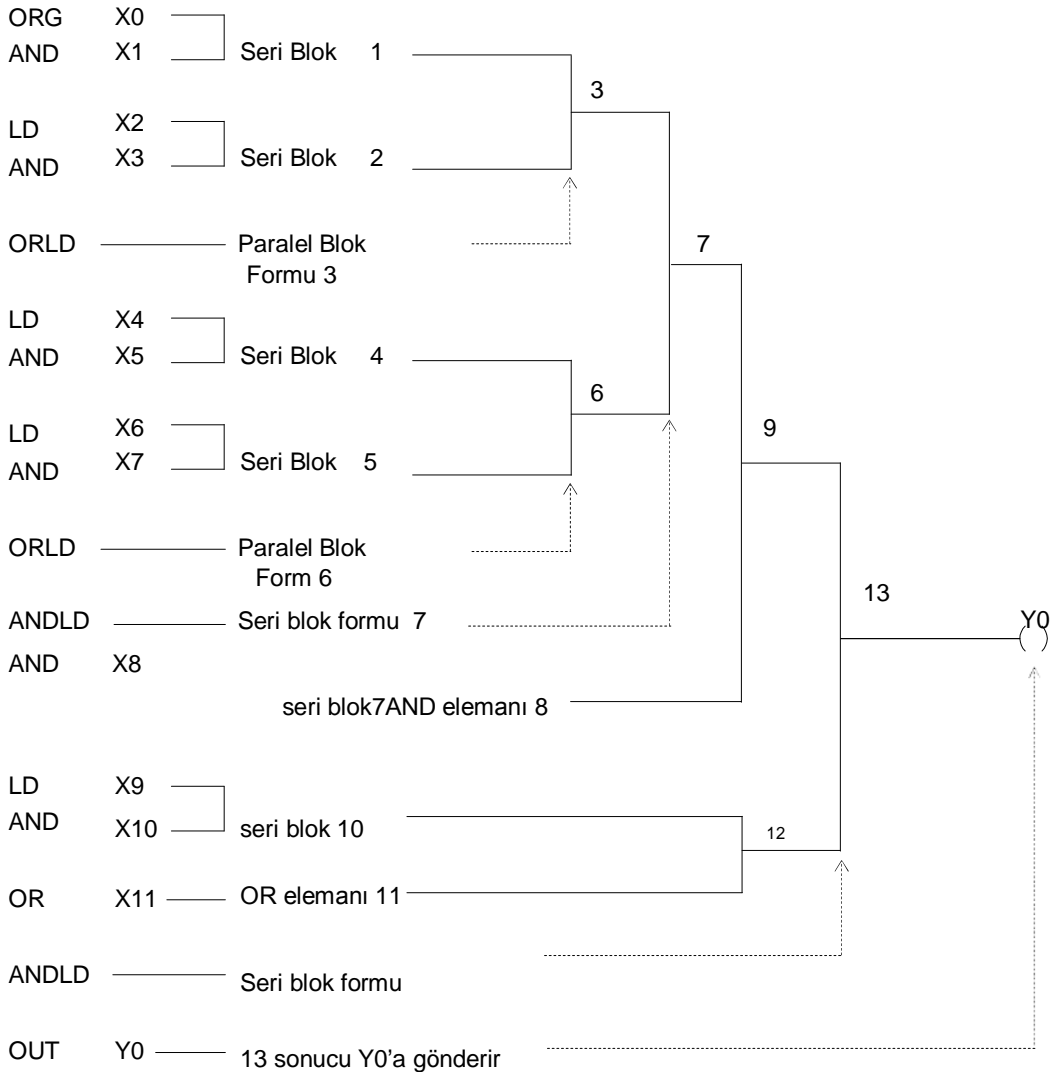
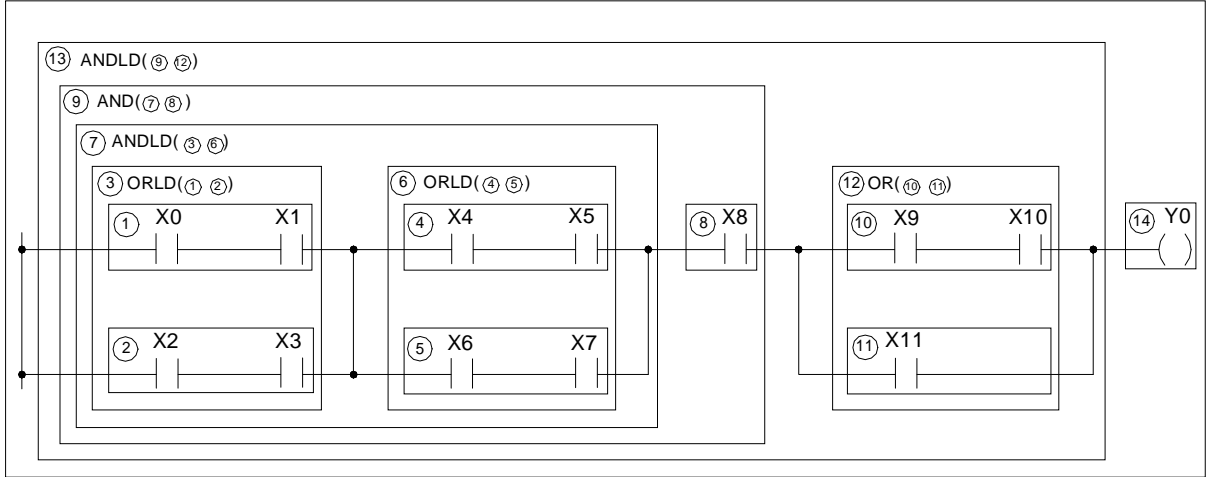
Y 0

1.5 Ağın Ayırışması

(WinProLadder kullananlar bu bölümü geçebilir)

Network'un de-compositon'ının tuş işlemi bağımsız elementlerin iki yatay çizgi arasında oluşan devreleri ve seri blokları ayırmak için, sonrada o elementleri kodlayarak ve mnemonic kodlama kurallarına göre seri bloklar ve daha sonra da sağdan sola aşağıdan yukarı şekilde seri halde bağlayarak (ANDLD yada ORLD komutunu kullanarak) yada seri paralel bloklar halinde çözümlmek için kullanılır. Bu olay sonunda tamamlanmış network halini alır.

Örnek Diyagram:



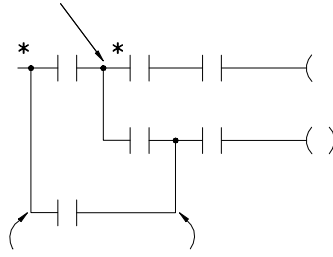
1.6 Geçici Rölelerin Kullanımı

(WinProLadder kullanıcıları bu bölümü geçebilir)

Bölüm 1,5'de gösterilmiş olan mnemonic kodlama için ağ çözümlene yöntemi dallanmış blok veya dallanmış devreye uygulanamaz. Bölüm 1,5'de gösterilen yöntem kullanılarak program girişi için ilk önce geçici rölelerin dallanmış noktalarının durumları depolanmalıdır. Program tasarımı, mümkün olduğu kadar dallanmış blok ve dallanmış devreden kaçınılmalıdır. İleriki bölümdeki "Program basitleştirme teknikleri" kısmına bakınız. İki durum içinde altta tanımlanan TR kullanılmalıdır.

- Dallanmış Devre: Birleştirilen hat, dal hattın sağ tarafında olamaz veya dal hattın sağ tarafında bir birleşme hattı vardır ama aynı satırda olamazlar.

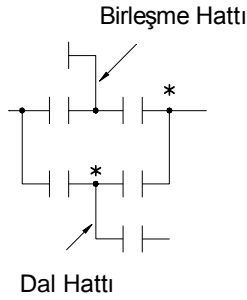
Örnek : * TR rölesinin ayarlarını gösterir
birleşik hatsız



Bu dal her ne kadar birleşme hatlarına sahipse de ama onlar aynı satırda değildir bu yüzden bu aynı zamanda bir dallanmış hattır.

- Dallanmış Blok : Blokların birinde dallı bir yatay paralel blok bulunmaktadır.

Örnek :



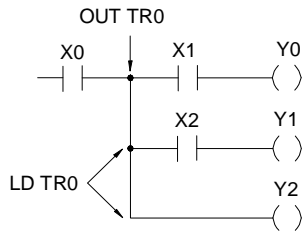
Açıklama 1: OUT TR komutu dallanmış noktanın üstünde programlanmış olmalıdır. LD TRn komutu devrelerin başlangıç noktalarında kullanılır. Dal hat durumunu yeniden kazanmak dal hattının ikinci satırları önce devrelere herhangi bir eleman bağlanabilmelidir. AND komutu OUT TRn veya LD TRn komutundan sonra ilk elemana bağlamada kullanılmalıdır. LD komutu bu durumda kullanılmaz.

Açıklama 2: Bir ağ 40 TR noktasına kadar sahiptir ve TR numarası aynı ağda defalarca kullanılmazlar. Dizilerin 1,2,3,... Şeklinde sıralı kullanılması önerilir. TR numarası aynı dal hattındaki ile aynı olmalıdır. Örneğin, eğer OUT TR0 kullanılan dal hattı, daha sonra 2. sıradan başlayarak LD TR0 bağlantısı için kullanılmalıdır.

Açıklama 3: Eğer dallanmış devre veya dallanmış blok dal başlangıç hattında ise daha sonra ORG veya LD komutları direk olarak kullanılabilirler ve TR kontağı gereksizdir.

Açıklama 4: Eğer dallanmış devre satırlarının herhangi biri çıkış bobinine (seri olarak bağlanmış elemanların arasında vardır) bağlanmamış ve diğer devreler ikinci sıradan sonraysa TR komutu dallı noktalarda kullanılmalıdır.

Örnek:

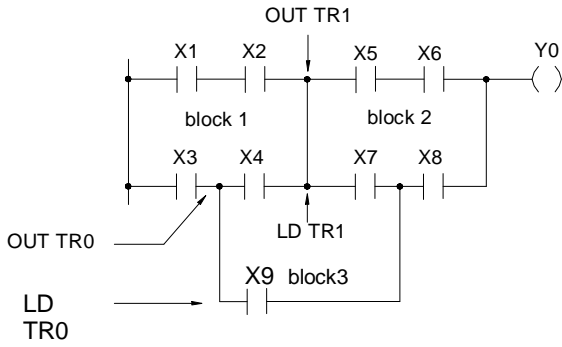


```

AND    X    0
OUT TR 0
AND    X    1
OUT    Y    0
LD TR  0 ← 2. sıradan başlar
AND    X    2
OUT    Y    1
LD TR  0 ← 3. sıradan başlar
OUT    Y    2

```

Örnek:



```

ORG    X    1
AND    X    2
LD     X    3
OUT TR 0
AND    X    4
ORLD
OUT TR 1
AND    X    5 ← AND komutu TR
AND    X    6 ← Komutundan sonra
LD TR 1 ← kullanılır.
AND    X    7 ← LD TR komutu TR
LD TR 0 ← branch hattına dönmek
AND    X    9 ← için kullanılır.
ORLD
AND    X    8
ORLD
OUT    Y    0

```

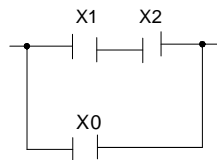
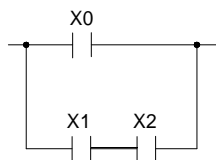
⌘ Seri olarak iki paralel bloğun bağlantı örneği yukarıda gösterilmektedir. X9 elemanı, dallanmış bloklar ve paralel bloklar ve ağ içinde tanımlandığında Blok 3 şekillenmiştir.

⌘ TR komutu gereksizdir çünkü (*) noktası başlangıç hattındadır.

⌘ Eğer iki blok seri olarak bağlanıp TR rölesi halen kullanılıyorsa, ANDLD komutu gereksizdir.

1.7 Program Basitleştirme Teknikleri

⌘ Eğer tek eleman seri bir bloğa paralel bağlanmışsa ve seri blok bu tek elemanın üstüne bağlanmışsa ORLD komutu eklenmelidir.



```

LD     X    0
LD     X    1
AND    X    2
ORLD

```

```

LD     X    1
AND    X    2
OR     X    0

```

⌘ Tek bir eleman veya seri blok paralel blokla paralel bağlanmışsa ve arkadaki paralel bloğa konmuşsa ANDLD komutu eklenmelidir.



```

ORG   X   0
AND   X   1
LD    X   2
LD    X   3
AND   X   4
ORLD
ANDLD

```

```

ORG   X   3
AND   X   4
OR    X   2
AND   X   0
AND   X   1

```

⌘ Eğer dal devresinin dal noktası çıkış bobinine direk olarak bağlanmışsa bu bobin indirgenen koda dal hattının üzerine (birinci satıra) yerleştirilmelidir.



```

OUT TR 0
AND   X   0
OUT   Y   0
LD TR 0
OUT   Y   1

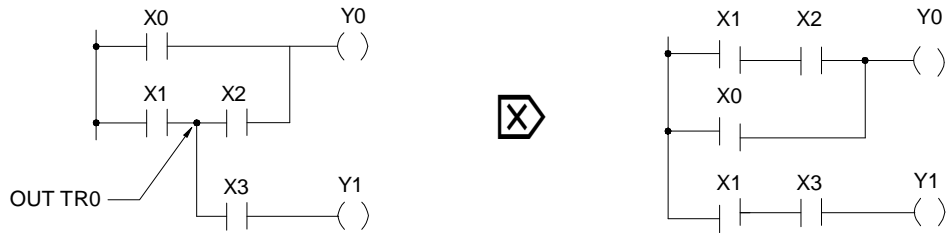
```

```

OUT   Y   1
AND   X   0
OUT   Y   0

```

⌘ Altta gösterilen diyagramda TR rölesi ve ORLD komutu eklenmelidir.



```

ORG   X   0
LD    X   1
OUT TR 0
AND   X   2
ORLD
OUT   Y   0
LD TR 0
AND   X   3
OUT   Y   1

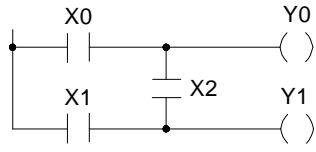
```

```

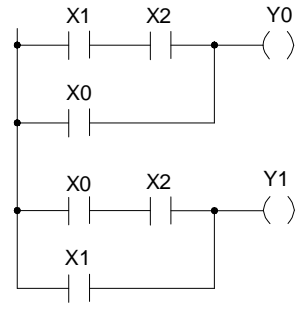
ORG   X   1
AND   X   2
OR    X   0
OUT   Y   0
ORG   X   1
AND   X   3
OUT   Y   1

```

⌘ Köprü devresinin dönüştürülmesi



Bu ağ yapısına PLC programında izin verilmemektedir.



```
ORG X 1
AND X 2
OR X 0
OUT Y 0
ORG X 0
AND X 2
OR X 1
OUT Y 1
```



Kısa Notlar